

embodiments, a single, contiguous data word of length x is partitioned into two smaller-length sub-sets, a and b , (where $x = a + b$), and then two independent and distinct EDACs operate in parallel to generate respective code words: in Fig. 1, $[(a + s) \& ((x - a) + t)]$; and in Fig. 2, $[(a + s) \& (b + t)]$. As explained in lines 3-6 on page 5, this arrangement reduces the complexity of a single, large EDAC by distributing the EDAC function across multiple, smaller EDAC circuits, thereby reducing the complexity of the EDAC calculation and increasing the speed of operation and throughput. Since there is nothing in Ozaki to suggest such an arrangement, Applicants are at a loss to understand the basis for the Examiner's rejection. Accordingly, Applicants respectfully request the Examiner to reconsider this rejection and to either withdraw the rejection or clearly indicate the perceived relevance of Ozaki to claims 1-9. Applicants wish to reserve the right to amend claims 10-13 so as to make them independent of claim 6 until such time as the allowability of claim 6 can be resolved.

In the Office Action, the Examiner has also rejected claims 14 and 15 under 35 U.S.C. § 103 as being obvious in view of Ozaki, but indicated the allowability of dependent claims 16-18. In response, Applicants have amended claim 14 to include all of the limitations of claim 16, amended claim 15 to reflect such amendment, and canceled claim 16. Accordingly, claims 14-15 and 17-23 should now be in condition for allowance.

In the Office Action, the Examiner has, in addition, rejected claims 24-30 under 35 U.S.C. § 103 as being obvious in view of Ozaki. While certainly relevant to the invention claimed in claims 24-30, Applicants must respectfully disagree with the Examiner that Ozaki renders obvious our invention for the purposes of 35 U.S.C. § 103. In Ozaki, the C2 parity generator 2 serially receives a set number of fixed-length data words, generates a C2 error correction code for each, and, with the assistance of the selector 3, stores the resultant (data + C2) codewords *by row* into the RAM 4 under control of the address controller 5. After this operation has been completed, then, during a second, completely independent operation, the C1 parity generator 6 cooperates with the address controller 5 to retrieve the bits of the stored codewords by column, generates the C1 error correction codes for each column, and, with the assistance of the selector 8, outputs the resulting doubly-encoded stream. Note that this stream never goes back into the RAM 4, nor does it ever get stored into the RAM 14 in the receiver -- in other words, the C1 codes are intended to be generated dynamically at the time of transmission and discarded shortly after reception. In the event that even a single data bit needs to be changed, the corresponding codeword must first be regenerated by the C2 parity generator 2 and, assuming the selector 3 and address controller 5 cooperate, the new codeword can then be stored in the RAM 4. However, since the C1 parity generator 6 operates dynamically, the **entire contents** of the RAM 4 must be retrieved in order to regenerate the correct C1 error correction codes! In contrast, Applicants bit-wise parity

generation circuitry operates on a cycle-by-cycle basis. As explained in lines 8-12 on page 11 of Applicants' specification:

This allows the parity indicator for each column to be updated at the time of each write to memory, based on the "old" bit" (*i.e.*, the respective bit of the code word previously written into that column), the "new" bit (*i.e.*, the respective bit of the new code word which is going to be written into that column), and the current value of the corresponding parity indicator.

Not only can our invention perform the parity generation (actually, regeneration) on a cycle-by-cycle basis, but it also allows error detection and correction on a cycle-by-cycle basis. As explained in lines 29-32 of page 14 of Applicants' specification:

In accordance with our invention, the BEDAC 54 effectively implements a classic "block error detection and correction" code, not in the traditional word-parallel manner, but bit-serially, *i.e.*, in whatever order the "data" bits happen to appear, including purely random.

To more particularly point out and distinctly claim this important difference, Applicants have amended claim 24 to indicate that the parity (either single or multiple bit) is performed each cycle in which any of the bits of an accessed row is stored (*i.e.*, during each write cycle). Since Ozaki cannot operate in such a manner, claim 24, and its dependent claims 25-30, now distinguish over this reference for the purposes of 35 U.S.C. § 103. Accordingly, claims 24-30 should now be in condition for allowance.

Finally, in the Office Action, the Examiner rejected claims 31 and 32 under 35 U.S.C. § 103 as being obvious in view of Ozaki. As explained above, Applicants respectfully submit that, in Ozaki, the C2 parity generator must receive the incoming data in a pre-determined sequence, while the C1 parity generator must receive the stored codewords in a different, but still pre-determined, sequence. In contrast, Applicants RAEDAC unit is specially adapted to receive data bits "in any order" (see, line 4 of claim 31), and to (re)generate the corresponding parity bits without have to reprocess the entire set of bits. Thus, unlike Ozaki which is fine for data that changes relatively infrequently, Applicants' invention is particularly well adapted for use in systems wherein the data changes frequently -- even cycle-by-cycle, which would render Ozaki inoperative (since it could simply not keep up with the changing data)! In any event, since Ozaki cannot operate in the claimed manner, claim 31, and its dependent claim 31, is clearly un-obvious for the purposes of 35 U.S.C. § 103. Accordingly, Applicants respectfully request the Examiner to reconsider and withdraw this rejection.

Since claims 32 and 33 cover primarily the simple parity form of the RAEDAC, Applicants have added new claims 33-35 to cover the full ECC version of the RAEDAC. Since neither Ozaki or any other reference of record

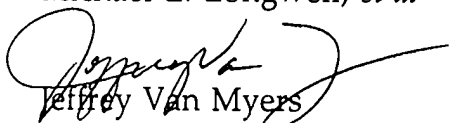
can operate in a random access mode, Applicants respectfully request the Examiner to consider and allow these new claims.

Conclusion:

Applicants respectfully request entry of the amendments proposed hereinabove. It is respectfully submitted that claims 1-15 and 17-32, as may be amended herein, and new claims 33-35 are allowable. In the belief that we have responded to each and every rejection contained in the Office Action of 27 September 2001, Applicants respectfully request the reconsideration and allowance of claims 1-15 and 17-35.

Respectfully submitted,

Michael L. Longwell, *et al*


Jeffrey Van Myers
Attorney for Applicants
Reg. No. 27,362
Ph: 512/858-7453

In the specification:

Replace the paragraph beginning on line 22 of page 6 with the following:

a1
In one embodiment of the present invention, a circuit for use in a memory system includes a memory having a plurality of dynamic memory cells arranged in a plane of rows and columns, each cell storing a bit, an access circuit connected to the memory to access, during an access sequence, all of the bits stored in said plane, and a row error detection circuit connected to the access circuit and said memory to detect an error in a bit of a row of said accessed bits. The row detection circuit having a parity generation circuit connected to said memory to generate a parity bit related to all bits stored in a respective one of said columns.

Replace the paragraph beginning on line 5 of page 16 with the following:

a2
By way of example, a typical T-flop, Tx, illustrated in Fig. 7, comprises a first D latch 64 and a second D latch 66, arranged in a classic toggle configuration with the Q output of the first D latch 64 coupled to the D input of the second D latch 66, and the D input of the first D latch 64 coupled to the Q output of the second D latch 66 via an inverter 68. An exclusive-Or gate (XOR gate 70) receives the new data bit to be stored in a memory cell of the DRAM 30, and the old data bit retrieved from that memory cell. So long as the old and new data bits match, the output of XOR gate 70 is a zero (0); when they differ, the output is a one (1). The output of XOR gate 70 is input into an OR gate 72, together with a write error signal. If either the old and new data values are different or the write error signal is asserted, the output of OR gate 72 is a one (1), and otherwise is a zero (0). As will be described below, the write error signal is used to correct the contents of the T-flop when an error has been detected in the old code word.

Replace the paragraph beginning on line 29 of page 16 and ending in line 6 of page 17 with the following:

R3 To facilitate system initialization, each D-flip-flop has an asynchronous initialization input, I, which is coupled to a Set/Clear (SET/CLEAR) signal, the logic state of which depends upon the location of the respective T-flop in the BEDAC 54, as shown, for the given example, in the left-most column in Fig. 5A. In this example, an odd parity scheme is being used so if an odd number of rows effect the BCBS signal the corresponding T-flop is cleared, and if an even number of rows effect the BCBS signal, the corresponding T-flop is set. In this way, the BCBS0 for each column is cleared, the BCBS1 for each column is cleared, *etc.* Since the initialization scheme will be determined at design time, the SET/CLEAR signal for each T-flop can be assigned the appropriate logic value in hardware.

In the claims:

14. (Once amended) A memory system comprising:

a memory comprising a plurality of dynamic memory cells arranged in a plurality of planes of rows and columns of said memory cells, each cell storing a bit and corresponding memory cells of each plane forming respective stacks;

an access circuit connected to the memory to access, during said access sequence, all of the bits stored in all of said planes of said memory cells; and

an orthogonal error detection circuit connected to the access circuit and said memory to detect an error in a bit accessed during said access sequence, comprising:

a row error detection circuit to detect an error in a bit of a row of said accessed bits;

a column error detection circuit to detect an error in a bit of a column of said accessed bits; and

a stack error detection circuit to detect an error in a bit of a stack of said accessed bits.

15. (Once amended) The memory system of claim 14 wherein said first, second and third error detection circuits also correct said bit errors, respectively.

14

24. (Once amended) A circuit for use in a memory system comprising:

a memory comprising a plurality of dynamic memory cells arranged in a plane of m rows and n columns, each cell storing a bit;

an access circuit connected to the memory to access, during an access cycle, all of the bits stored in a selected one of said rows; and

a row error detection circuit connected to the access circuit and said memory to detect an error in a bit of said row of said accessed bits;

the circuit comprising:

a parity generation circuit connected to said memory to generate, during each access cycle in which any of said accessed bits are stored, n parity bits, each related to all m bits stored in respective one of said columns.

31. (Once amended) A random access error detection and correction (RAEDAC) unit for detecting and correcting errors in an ordered bit string of predetermined length, the RAEDAC comprising:

a parity generation circuit which receives, in any order, each bit of said string, and generates a plurality of parity bits, each related to a unique combination of said bits comprising said string;

a parity check circuit connected to said parity generation circuit to detect an error in a bit of said string using said parity bit; and

an error correction circuit coupled to the parity check circuit to correct said detected bit error.

32. (Once amended) The RAEDAC of claim 31 wherein the parity check circuit detects multi-bit errors in said string using said parity bits.

33. (New) A random access error detection and correction (RAEDAC) unit for detecting and correcting errors in an ordered bit string of predetermined length, the RAEDAC comprising:

an error correction code generation circuit which receives, in any order, each bit of said string, and generates a plurality of check bits, each related to a unique combination of said bits comprising said string; and

an error detection circuit connected to said error correction code generation circuit to detect an error in a bit of said string using said check bits.

34. (New) The RAEDAC of claim 33 further comprising:

an error correction circuit coupled to the error detection circuit to correct said detected bit error.

35. (New) The RAEDAC of claim 33 wherein the error detection circuit detects multi-bit errors in said string using said check bits.